

M2 MPRI - Programmation Probabiliste - Discrete semantics

Christine Tasson

Exercice 1. *Equations*

Démontrer que les programmes suivants ont la même sémantique en vous aidant de la Figure 1

<pre>def coin(): x = sample(Bernoulli(0.5)) return x</pre>	\equiv	<pre>def coin(): x = sample(Binomial(2n, 0.5)) return x > n</pre>
<hr/>		
<pre>assume(a); ... ; assume(b)</pre>	\equiv	<pre>... ; assume (a and b)</pre>
<hr/>		
<pre>x = sample(d); y = sample(d')</pre>	\equiv	<pre>y = sample(d'); x = sample(d)</pre>

Exercice 2. *Coin*

Calculer la sémantique des programmes suivants.

```
def flip(p):  
    x = sample(Bernoulli(p), name='x')  
    y = sample(Bernoulli(p), name='y')  
    assume(x!=y)  
    return x  
  
def stop(p:float):  
    n = 0  
    while sample(Bernoulli(p)):  
        n = n+1  
    return n
```

Syntaxe :

$$\begin{aligned}
t &::= \text{None} \mid \text{bool} \mid \text{int} \mid t \times t \mid \text{dist}(t) \\
e &::= c \mid x \mid (e, e) \mid \text{op}(e) \\
s &::= \text{pass} \mid x = e \mid x = f(e) \mid \text{if } e : s \text{ else } : s \mid s ; s \mid x = \text{sample}(e) \mid \text{assume}(e) \\
d &::= \text{def } f(x) : s \text{ return } e \mid d \ d
\end{aligned}$$

Types :

$\llbracket t \rrbracket$	$= t$	Countable set
$\llbracket \text{None} \rrbracket$	$= \{*\}$	one point set
$\llbracket \text{bool} \rrbracket$	$= \mathbb{B}$	two points <i>True</i> et <i>False</i> sets
$\llbracket \text{int} \rrbracket$	$= \mathbb{N}$	integers
$\llbracket t_1 \times t_2 \rrbracket$	$= t_1 \times t_2$	products
$\llbracket \text{dist}(t) \rrbracket$	$= t \rightarrow \mathbb{R}^+$	value to score tables

Expressions :

$$\begin{aligned}
\llbracket c \rrbracket_\gamma^\varphi &= c \\
\llbracket x \rrbracket_\gamma^\varphi &= \gamma(x) \\
\llbracket \text{op}(e) \rrbracket_\gamma^\varphi &= \text{op}(\llbracket e \rrbracket_\gamma^\varphi) \\
\llbracket (e_1, e_2) \rrbracket_\gamma^\varphi &= (\llbracket e_1 \rrbracket_\gamma^\varphi, \llbracket e_2 \rrbracket_\gamma^\varphi)
\end{aligned}$$

Commandes :

$$\begin{aligned}
\llbracket \text{pass} \rrbracket_\gamma^\varphi(\gamma') &= \delta_\gamma(\gamma') = 1 \text{ if } \gamma = \gamma' \text{ and } 0 \text{ otherwise} \\
\llbracket x = e \rrbracket_\gamma^\varphi(\gamma') &= \delta_{\gamma + [x \leftarrow \llbracket e \rrbracket_\gamma^\varphi]}(\gamma') \\
\llbracket \text{if } e : s_1 \text{ else } : s_2 \rrbracket_\gamma^\varphi(\gamma') &= \llbracket s_1 \rrbracket_\gamma^\varphi(\gamma') \text{ if } \llbracket e \rrbracket_\gamma^\varphi \text{ otherwise } \llbracket s_2 \rrbracket_\gamma^\varphi(\gamma') \\
\llbracket x = \text{sample}(e) \rrbracket_\gamma^\varphi(\gamma') &= \sum_v \llbracket e \rrbracket_\gamma^\varphi(v) \delta_{\gamma + [x \leftarrow v]}(\gamma') \\
\llbracket x = f(e) \rrbracket_\gamma^\varphi(\gamma') &= \sum_v \mu(v) \delta_{\gamma + [x \leftarrow v]}(\gamma') \text{ with } \mu = \varphi(f)(\llbracket e \rrbracket_\gamma^\varphi) \\
\llbracket \text{assume}(e) \rrbracket_\gamma^\varphi(\gamma') &= \delta_{\text{True}}(\llbracket e \rrbracket_\gamma^\varphi) \delta_\gamma(\gamma') \\
\llbracket s_1 ; s_2 \rrbracket_\gamma^\varphi(\gamma') &= \sum_{\gamma_1} \llbracket s_1 \rrbracket_{\gamma_1}^\varphi(\gamma_1) \llbracket s_2 \rrbracket_{\gamma_1}^\varphi(\gamma')
\end{aligned}$$

Déclarations et inférence :

$$\begin{aligned}
\llbracket \text{def } f(x) : s \text{ return } e \rrbracket^\varphi &= \varphi + [f \leftarrow \lambda v. \lambda v'. \theta(v, v')] \\
&\quad \text{with } \theta(v, v') = \sum_{\gamma} \llbracket s \rrbracket_{[x \leftarrow v]}^\varphi(\gamma) \delta_{\llbracket e \rrbracket_\gamma^\varphi}(v') \\
\llbracket d_1 \ d_2 \rrbracket^\varphi &= \llbracket d_2 \rrbracket^{\varphi_1} \text{ with } \varphi_1 = \llbracket d_1 \rrbracket^\varphi \\
\llbracket \text{infer}(f, e) \rrbracket_\gamma^\varphi(v) &= \begin{cases} \frac{\mu(v)}{\mu(\top)} & \text{if } 0 < \mu(\top) < \infty \\ \text{Error} & \text{otherwise} \end{cases} \\
&\quad \text{with } \mu(v) = \varphi(f)(\llbracket e \rrbracket_\gamma^\varphi) \\
&\quad \text{and } \mu(\top) = \sum_v \mu(v)
\end{aligned}$$

Figure 1 : Sémantique de pIMP.